Institute of Software Technology

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Special Research Software Engineering No. 224

Evaluation of Application Performance Management Tools in the Context of an Existing Enterprise IT Landscape

Jakob Benz, Matteo Sassano, Valentin Seifermann

Course of Study:	Softwaretechnik B.Sc.
Examiner:	Dr.-Ing. André van Hoorn
Supervisor:	Dr.-Ing. André van Hoorn
Commenced:	2015/08/03
Completed:	2016/02/02
CR-Classification:	C.4

Abstract

Today's enterprises and especially their IT landscapes differ dramatically from the ones back in the 20th century. You could say that enterprises nowadays are some kind of "application-driven". With the IT infrastructure in today's globalized world moving to the cloud and at the same time getting way larger and more complex, Application Performance Monitoring has been a subject in many debates. Application Performance Management Tools therefore became an integral part of every (big) enterprise IT landscape.

This special research evaluates two tools of APM market leaders. The overall goal is to determine which of the APM tools fits better for the IT landscape of Industry Partner 1 and Industry Partner 2 . Due to privacy issues, the industry partners and APM tools used in this study are anonymized. The solutions of Vendor 1 and Vendor 2 therefore are compared within two aspects:

- APM Use-Cases - evaluation by means of classic APM disciplins, and
- APP Use-Cases - real scenarios in the context of 3 applications provided by Industry Partner 1 and Industry Partner 2

The assessment is done in cooperation with Industry Partner 1 and Industry Partner 2 representatives and will be presented in an "aggregated view" in the end.

Zusammenfassung

Die heutige Unternehmenslandschaft und vor allem die IT-Systeme der (Produktions-) Unternehmen haben sich im Vergleich zum vergangenen Jahrhundert stark verändert. Man könnte sogar sagen, dass Unternehmen "Applikations-getrieben" sind. Während sich die IT Infrastruktur in der globalisierten Welt immer mehr in Richtung Cloud bewegt und gleichzeitig größer und komplexer wird, werden Application Performance Monitoring Tools ein integraler und unverzichtbarer Teil einer jeden (größeren) IT Landschaft.

In dieser Fachstudie werden die Tools zweier APM Marktführer verglichen. Das Ziel dieser Studie ist der Vergleich dieser beiden Tools im produktiven Applikationsbetrieb von Industriepartner 1 und Industriepartner 2, um anschließend eine Empfehlung geben zu können. Es wird anhand von zwei größeren Kategorien verglichen:

- APM Use-Cases - Evaluierung im Sinne von klassischen APM-Disziplinen, und
- APP Use-Cases - Evaluierung anhand von realistischen (Einsatz-) Szenarien in den 3 von Industriepartner 1 und Industriepartner 2 bereitgestellten Applikationen.

Die Bewertung der einzelnen Use-Cases findet in Kooperation mit Vertretern von Industriepartner 1 und Industriepartner 2 statt, und wird in einer aggregierten Tabelle am Ende präsentiert.

Contents

1 Introduction 1
1.1 Motivation 1
1.2 Industry Partner 1
1.3 Goals 1
1.4 Document structure 2

2 Course of the study 3
2.1 Timeline 3
2.2 Phases 3
2.3 Milestones 5

3 APM-Tools and Applications 7
3.1 What is Application Performance Monitoring 7
3.2 Application-Performance-Management-Tools 7
3.3 Applications 8

4 APM Use-Cases 9
4.1 Use Cases 9
4.2 Use-Case-Evaluation 16

5 Application Use-Cases 55
5.1 Use-Cases 55
5.2 Use-Case-Evaluation 55

6 Results & Overview 59
6.1 Assessment Overview 60
6.2 Overall Conclusion 61
6.3 Recommendation 61

Bibliography 63

Chapter 1

Introduction

1.1 Motivation

Today's enterprises are using many complex software systems in order to be able to manage highly automated production (processes) and to maintain support for millions of customers. If., e.g., the response time of a shopping website is just a little bit too slow, customers won't wait, but switch to another vendor, which means less sales for the enterprise. Monitoring software systems (especially high-availability-software) in today's enterprises is therefore indispensable.

There is a wide a range of Application Performance Management Tools available on the market, going from free open source software to much more extensive and expensive tools. Large enterprises are experiencing troubles, when they are trying to evaluate the most suitable Application Performance Monitoring Tool. This is why the evaluation process needs to be done by an independant group of experts, in this case — we three students.

1.2 Industry Partner

This study is conducted in cooperation with Industry Partner 1 and Industry Partner 2.

1.3 Goals

The overall goal of the study is to give a recommendation, which tool fits best for the industry partners' IT landscape.

The decision shall be based on both a theoretical contrasting juxtaposition and the evaluation in productive application operations. Vendor 1 and Vendor 2 therefore are compared within Application-Performance-Management-oriented as well as Application-oriented use cases.

Furthermore, both solutions are examined by representatives of each application, the feedback of those people will be considered during documentation and assessment.

1.4 Document structure

Following this, Chapter 2 gives an overview of the timetable and the milestones of the study as well as a brief description on how we approached the different tasks. Chapter 3 presents both the APM Tools of Vendor 1 and Vendor 2 and the applications by Industry Partner 1 and Industry Partner 2 used for the evaluation in the productive application operations. The APM use cases are listed with a short description and a set of evaluation criteria in Chapter 4. This Chapter also includes the documentation and evaluation of the APM use cases. After that, in Chapter 5, the scenarios (APP use cases) are presented and documented. Finally, Chapter 6 includes the assessment overview, the overall and final conclusion as well as the recommendation.

Chapter 2

Course of the study

This chapter will describe the course of the study, which was conducted over six/seven months in the period between 3rd August 2015 and 2nd February 2016.

2.1 Timeline

The diagram below shows the phases and the reached milestones of the study.

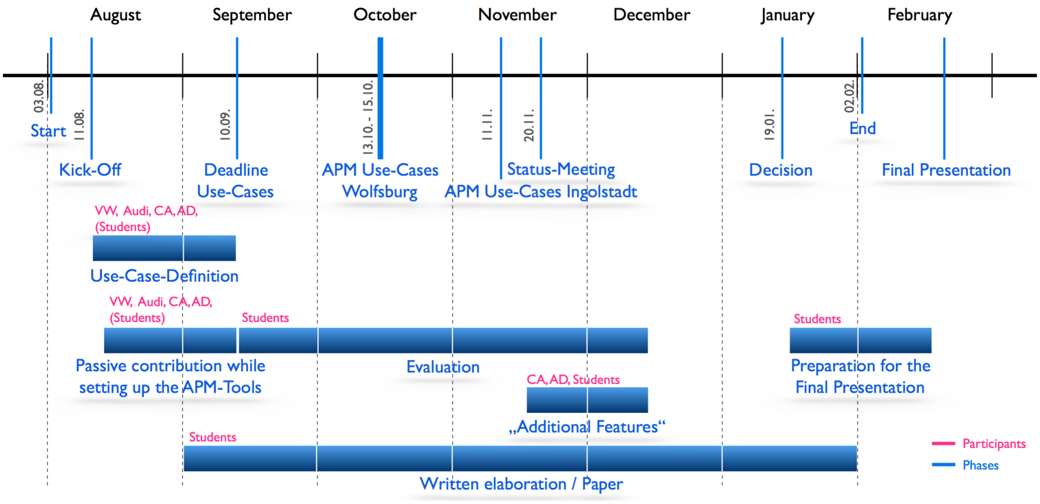


Figure 2.1. Timeline

2.2 Phases

2.2.1 Phase 1: Kick-Off

The study began with the kickoff meeting on 11th August 2015 in the department of our industry partner Industry Partner 2 in Ingolstadt. Industry Partner 1 and Industry

Partner 2 presented the participant and presented the task for this study. Subsequently we presented our slides where we introduced our course, goals, schedule and structure of the study. At the end of the meeting the Applications, which are going to be instrumented, were presented from their responsible.

2.2.2 Phase 2: Introduction to the APM-Tools

In fact that we were new in this topic, as well as Vendor 1 introduced us in the APM World, presented us their solutions and explained how they work. Later both of them gave us access to demo environments of their products.

2.2.3 Phase 3: Definition of the use cases

We started to garner ideas for criteria that an APM Tool has to accomplish. The application responsible made us clear what points are important for them, and with their help we finished our first use cases document which later was approved with some small changes from Industry Partner 2 and Industry Partner 1 .

2.2.4 Phase 4: APM Tools Setup

For monitoring the above listed systems, the APM Tools had to be set up. To evaluate and compare the installation and the setup, we took passively part in some installation sessions on-side and via video conference.

2.2.5 Phase 5: Evaluation

Phase 5.1: Evaluation - APM use cases Wolfsburg (System A)

The evaluation started in Wolfsburg. We could compare the two solutions side by side and we could ask the vendors directly some important and relevant questions for our study. In this 3 days long session we were able to complete evaluating most of the use cases.

Phase 5.2: Evaluation - APM use cases Ingolstadt (System C and System B)

The focus in this session was to research the missing use cases and optimization. With optimization we wanted to compare the solutions of Vendor 1 and Vendor 2 in a more practical way. Therefore we listed up 5 new use cases which were dependent to the monitored application.

2.2.6 Phase 6: Tool recommendation

For the tool recommendation, we had to analyze the results of the evaluation. We granted points to both vendors for each use case. At the end, we summarized all collected information and wrote an overall conclusion.

2.3 Milestones

2.3.1 Milestone 1: Use cases deadline

10th September 2015:
For this milestone we had to gather information about APM and think about criteria important for an APM Tool. At least we had to summarize them as use cases.

Milestone result

Finished use case document

2.3.2 Milestone 2: Evaluation and conclusions

6th January 2016:
The goals of the second milestone were to do the evaluation and to document to every use case what Vendor 1 and Vendor 2 were able to do. In addition to that we made conclusions to every use case. Another goal of this milestone was to begin the documentation.

Milestone result

Unfinished paper

2.3.3 Milestone 3: Paper and Presentation

2nd February 2016:
For the last milestone of our study we wanted have a finished documentation and and second one was to prepare the final presentation

Milestone result

Finished Paper

Chapter 3

APM-Tools and Applications

3.1 What is Application Performance Monitoring

"'Gartner defines application performance monitoring (APM) as one or more software and hardware components that facilitate monitoring to meet five main functional dimensions: end-user experience monitoring (EUM), runtime application architecture discovery modeling and display, user-defined trans- action profiling, component deep-dive monitoring in application context, and analytics."' *[Gartner IT Glossary - Application Performance Monitoring (APM)]*

3.2 Application-Performance-Management-Tools

3.2.1 Vendor 1

Vendor 1 was founded in New York in year 1976 under the name Computer Associates International, Inc. In 2004, the company was renamed. This company is one of the biggest software companies in the world which sells management software for networks, storage, security, and project management. The APM module is only one of many software solutions offered by Vendor 1 . Other products are ,e.g., Infrastructure Management, Service Management and Workload Automation tools. Furthermore Vendor 1 offers solutions for the Mainframe and Security Sector.

3.2.2 Vendor 2

In 2008, Vendor 2 was founded under the name Singularity. This company received its actual name in year 2010. The Board of Directors of Vendor 2 consists of experienced innovators with huge experience in the fields of Application Performance Management, Systems Management and Distributed Systems. They were active at Vendor 1 , Adobe, HP Mercury, BMC, Oracle, Sun and McAfee and already have developed solutions for thousands of organizations. The company focuses on managing the performance and availability of applications.

3.3 Applications

3.3.1 System A

System A is an IT system within a modul familiy specifically developped for fine tuning based on inspection technology. By using a guideline of correct tolerance values the system does an inspection of components as well as their composition. It needs to be ensured that all components are build in validly. This will be done by comparing the ID numbers of components. At the assembly the system also heads the necessary data to program the controller components individually. Overall, the landscape consists of databases, Servers and clients and inspection systems. The communication is performed via HTTPS.

3.3.2 System B

System B is a web application, which is embedded in a website. In this system a list of selectable components is available. With the selected components and favorites a desired product can be build. In addition, it is possible to generate certain codes and build a PDF-file.

3.3.3 System C

System C is a website. Web applications are using a web browser as a client to update and maintain web applications without distributing and installing software on potentially thousands of client computers. In this system you can see new products and get a lot of information about them. This system is publicly accessible.

Chapter 4

APM Use-Cases

4.1 Use Cases

In Chapter 4.1, the use cases used for the analysis will be presented. This includes also the considered evaluation criteria. Some of the following aspects are not measurable in a 100% objective and quantitative way. Nonetheless, it should be mindful of using a transparent, quantitative, as well as use case specified evaluation dial during the assessment. This also applies to all possible operational procedures in today's enterprise environments and the development of monitored software.

The use cases are grouped in five major sections. Each group of use cases has a priority $Prio \in \{1, 2, 3\}$, where 1 is the highest and 3 the lowest priority. During the assessment, each single use case will be graded with a score, where 0 points implies full satisfaction of the use case's criteria. Additional points will be given / subtracted, if there are features on top / deficits. In the end, there will be an aggregated view presenting the overall results.

4.1.1 Installation

This group of use cases has a priority of 2.

APM-UC-01 - Installation

With this use case, the effort and the simplicity of the installation of both APM solutions in an existing IT landscape shall be evaluated. Therefore, the following evaluation criteria need to be considered:

- Effort of the preparation in advance of installation by the costumer
- Complexity and duration of the installation process in consideration of the complexity of the monitored system
- Supported languages (e.g., Java, .NET, etc.)
- Required staff and know-how
- Required hardware and software

- Any additional configuration of the agents
- Used products to fullfil monitoring

APM-UC-2 - User Management

Here, the possibility of the dynamic user management relating to the usage of controllers will be vetted. Therefore, the following evaluation criteria need to be considered:

- User permissions and definitions
- User groups
- Role systems (experts, contacts, etc.)

APM-UC-3 - Integration / Extensibility

The integration in systems of the existing IT Landscape in todays enterprises as well as the extensibility of the APM solution itself will be considered in this use case. Therefore, the following evaluation criteria need to be considered:

- Possibility for access to the collected data (e.g., by REST API)
- Integration possibilities with
  - Infrastructure management tools
  - DevOps tools
  - Reporting systems
  - Load drivers und simulators

4.1.2 Configuration

This group of use cases has a priority of 1.

APM-UC-04 - Status-Quo Identification

Here, the focus lies on the identification of the actual state of the the monitored system in regard to performance problems and rescource management, just after installing the APM solutions. Therefore, the following evaluation criteria need to be considered:

- Duration until the earliest results exist,
- Representativity of those results,
- Significance and
- Scope of the results

APM-UC-05 - Runtime Application Architecture

The second of Gartner's 5 Dimensions of APM is examined in this use case. The possibility to describe the underlying components and diagnose the overall system state. Therefore, the following evaluation criteria need to be considered:

- Application discovery & dependency mapping (e.g., in a SOA)
- Distributed monitoring
- Correlation of communication

APM-UC-06 - Visualization / Dashboards

The visualization and preparation of the received data as well as the grade of intuitiveness is very important in respect of the usability. Here, the general display of the user interface and also the possibility to create dashboards and configuration will be vetted. In addition to that, the possibilities of user defined transaction profiling (third of Gartner's 5 Dimensions of APM) will be examined. Therefore, the following evaluation criteria need to be considered:

- Flexibility
- Clarity
- Simplicity
- Intuitiveness
- Visualization of different drill downs
- Performance of the APM solution itself
- Dashboards - automated / manual configuration, storage, Import / Export
- Differentiation between user groups (1st-Level-Support, Power-User, etc.)
- Release management

APM-UC-07 - Agent Configuration

In this use case the configuration possibilities of the agents will be evaluated during operation. Therefore, the following evaluation criteria need to be considered:

- Possibility and range of the configuration
- Restrictions
- Effects on the running application

- Reusability of configurations
- Preconfigured / optimized agents (e.g., Web-Server, Databases, etc.)
- Effort for the configuration during updates at the middleware (Tomcat etc.)

4.1.3 Daily use

This group of use cases has a priority of 1.

APM-UC-08 - Hardware Monitoring

This use case takes a closer look at the possibility to monitor and analize the

- Usage,
- Workload and
- Availability

of available resscources, e.g., CPU, Memory, Ressource-Pools etc. Therefore, the following evaluation criteria need to be considered:

- Available metrics
- Understandability

APM-UC-09 - Root Cause Analysis / Deep Diving

In this use case the question "Where is the error?" has priority. The possibility to identify the root cause of already recognized performance problems as well as the tracing shall be considered closely. This relates to the fourth of Gartner's 5 Dimensions of APM. Therefore, the following evaluation criteria need to be considered:

- Identification of
    - slow methods,
    - hanging threads,
    - unperformant SQL-Statements & response time,
    - render time and
    - messages,
- Available metrics,
- Filter possibilities,

- Error information (when did the error occur for the first time),
- Tracing (continuously, automaticaly, manually configurable, availability),
- Drill Down,
- (Transaction) Snapshots,
- Business Transaction Analysis,
- Triage,
- Client monitoring,
- Database monitoring and
- Analysis of unstructured data.

APM-UC-10 - Analytics

Under this point, the possibility of analizing the collected data will be evaluated. Especially the comparison of two or more problem / error situations as well as the detection of trends. Therefore, the following evaluation criteria need to be considered:

- Baselines - automatic / manuel configuration,
- Trends (e.g., daily, end-of-business-hour, before / after release / deployment etc.),
  - Comparison of expectancy values and
  - Identification of vulnerabilities within the architecture or code,
- Application cluster and
- Import / export function

APM-UC-11 - Pro-Active Monitoring

Now the possibility of "pro-active" monitoring will be considered. Automated reactions of well-defined performance problems as well as creating a prediction of potential problem situations are central. Therefore, the following evaluation criteria need to be considered:

- Thresholds,
  - On which value thresholds can be set,
  - Automatic/ manual configuration and
  - Definition of tresholds to create an alert relating to business transactions as well as technical tresholds,

- Alerting,
  - Different reaction possibilities / notifications and
  - Identification of possible contacts / experts, and
- Alert-Management,
  - Alart-rules and
  - Dependencies and logical links of any alerts.

APM-UC-12 - Automation

In this use case the automation in regard to workflows — for the purpose of pro-active monitoring — will be evaluated closely. Therefore, the following evaluation criteria need to be considered:

- "Sophisticated Reactions" such as error treatment scripts, configuration of a machine and other automatable actions, and
- Possibilities of creating templates for automated workflows.

4.1.4 Agent Functionality

This group of use cases has a priority of 2.

APM-UC-13 - Scalability

In this use case the scalability of the APM solution will be vetted. Therefore, the following evaluation criteria need to be considered:

- Maximum number of agents (per controller), and
- Requirements on the hardware (to save data)

APM-UC-14 - Agent Functionality / Data Collection

The way the agents collect data from each particular app-server, machine, database etc. differs in many ways. Most notably both data collection strategies shall be presented in this use case. The most important aspect to be considered is the overhead the agent produces during "work".

4.1.5 Privacy

This group of use cases has a priority of 2.

APM-UC-15 - Privacy

In this use case the data privacy of the solution will be considered precisely. At this, the focus will be set on the collection, transmission and storage of data in general as well as sensitive, personal or worth protecting information. Therefore, the following evaluation criteria need to be considered:

- Which data will be colected according to data privacy,
- Storage of sensitive data (protection, encryption),
- Transmission of sensitive data (protection, encryption),
- Configuration possibilities and
- Certification under security standards

4.2 Use-Case-Evaluation

4.2.1 APM-UC-01 - Installation

Vendor 1

The installation of the Vendor 1 solution can be done by placing an already provided zip file on the application server. The agents can be downloaded seperatly from their website. To fulfill monitoring the Enterprise Manager needs to be installed. This manager receives all information sent by the agents. The customer can choose between a standalone Enterprise Manager or a cluster landscape. Therefore, many Enterprise Managers can be installed. At least one Enterprise Manager must be set as an MoM (Manager of Manager) to be the head component of a cluster. After downloading and placing the agent data on the application server, the agent will be installed by using an installation wizzard. During this process, different configruations will be set. This refers also to the Enterprise Manager. In general, the agent directory path has to be configured manually. As mentioned before, during the installation, it is possible to adjust server and application configurations. Therefore, server and application know-how is recommended. In reference to the configuration, extensions are possible. Vendor 1 applies three different tools to fulfill monitoring:

- TeamCenter
- WebView
- Fat-Client Workstation

In general, Vendor 1 offers a wide spectrum of agents:

- Browser Agent
- EP Agent
- Java Agent
- NET Agent
- Node.js Agent
- PHP Agent
- Mobile App Analytics

Vendor 2

The installation of Vendor 2 in the System A environment is quite easy. Besides the preparation of the hardware (e.g., checking rights and permissions), the customer has nothing to do in the run-up of the actual installation. It may be the case that a server admin or a person responsable for the monitored application is needed. During the installation of the Vendor 2 tool, 1 server-admin and 1 manager of System A had been consulted. The installation of the controller is simply done by copying the software package to the specified server. The agents are available for download from a web interface, where prior to the that the initial agent configuration is done (programming language etc.). If the configuration of an agent is done through this very web interface, the `controller-info.xml`-file is automatically created and filled out. System A is implemented in Java, therefore the `-javaagent`-paramater had to be added to the JVM arguments, which specifies the directory of the agent. The agent then comes out-of-the-box and will immediately start to report to the controller after start-up. Generally, agents are, among others, available for Java, .NET, PHP, .NodeJS, Python, C/C++. Furthermore, machine agents, agents for client monitoring and browser monitoring, iOS / Android are available. Excluding the time used for the preparation on the part of Industry Partner 1 and waiting for the server admin, the whole installation process took about half an hour. Note, that this only represents the installation of the controller and some single agents, not a roll-out of, e.g., 1000 or even more agents.

Conclusion

The effort and simplicity were the two main criteria, we wanted to evaluate with this use case. Both environments were, with some small differences, quite simple to install for perople with certain know-how. Vendor 2 has an advantage, when it comes to the duration of the agent configuration. Vendor 2 offers Python and C/C++ agents, Vendor 1 does not.

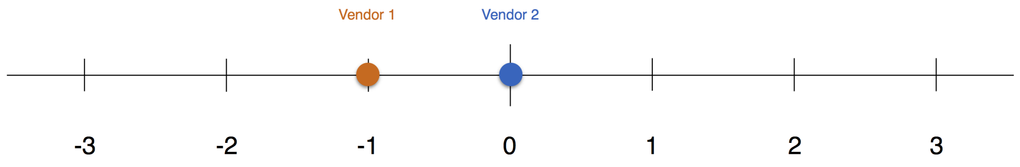


Figure 4.1. Assessment of use case no. 1

4.2.2 APM-UC-02 - User Management

Vendor 1

In consideration of the size of the enterprise, sometimes variable access to the data is necessary. Vendor 1 solution does not offer user management. In the tool itself it is only possible to indicate one responsible person for any component manually. To get access to the data and dashboards, the user must have all the login data.

Vendor 2

Vendor 2 user management offers all basic functionality like adding / editing / deleting users. The role-based access control system lets the user manage the permissions of users at a very fine-grained level, e.g., the user can

- give permissions for only one application / tier / node, and
- give permissions for only a specific dashboard.

Vendor 2 ships with some predefined roles like, e.g., "Account Owner", "Administrator", "Workflow Executer", "Read Only User" etc. Furthermore, the user can define custom roles which also can be inherited from other existing roles. Additionally, the user can assign an expert role to users, thereby, you can configure an alert to automatically contact the expert for, e.g., an application.

Vendor 2 supports the following authentication providers: Vendor 2 itself, LDAP and SAML.

Conclusion

Vendor 1 ′s solution does not offer user management. They are offering an additional module, which provides user management, but we have to mention that it is not part of APM. To have access to data and dashboards, all the users must be registered. Vendor 2 offers user management out of the box. They offer predefined roles and give the opportunity to create custom ones. In the application of Vendor 2 the user can even share things like dashboards with users not registered in the system.

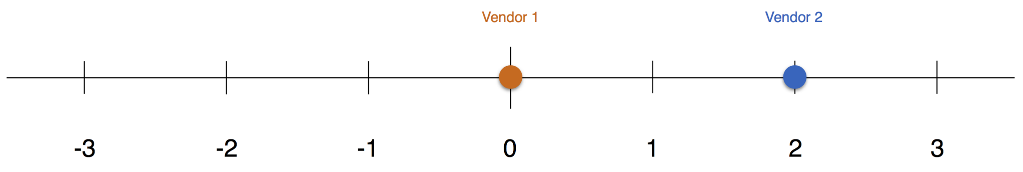


Figure 4.2. Assessment of use case no. 2

4.2.3 APM-UC-03 - Integration / Extensibility

Vendor 1

In general, Vendor 1 provides a REST API for users to have the possibility to get access to the collected data. Therefore, Vendor 1 supports the following public RESTful web services:

- Root Resource
- Vertex
- Vertex id

In general, the following data is accessible:

- alert / alert definition
- metrics data of all available metrics
- list of agents
- list of metrics paths
- dashboards
- application, servers, IP-address information

Vendor 2

Vendor 2 is providing a REST API to enable users to retrieve information programmatically. Retrievable data can be

- Information about applications, servers, IP-adresses (meta data),
- All metrics available,
- Customer data (e.g., sales, ratings, etc.) (can be configured),
- Every health status and
- Every event.

Besides retrieving data out of the controller, the user also can configure controller settings via the REST API.

To simplify and accelerate the access on data or even dashboards displayed in the controller, Vendor 2 offers links to those sites in the controller, which can be sent in, e.g., a notification sent due to a Health Rule Violation.

With HTTP request actions and templates the user is able to react to events triggered by a policy, in order to integrate a third-party HTTP API. HTTP request templates facilitate defining a HTTP request by providing a graphical interface.

In terms of extensibility, Vendor 2 offers existing extensions and SDKs for developing

- Monitoring extensions,
- Alerting extensions and
- Cloud auto-scaling extensions.

Custom extensions can be written in any programming language. Vendor 2 provides support for custom extensions, too. An example for a custom extension is the Jenkins Vendor 2 Plugin. It integrates data fetched from the REST API into a Jenkins build.

Conclusion

With this use case we wanted to explore what possibilities both solutions have for accessing to collected data and for extending the APM solution. In general both vendors are providing a REST API to enable users to obtain information of collected data. Both of them provide access to:

- all metrics
- dashboards
- and alert status

In the solution of Vendor 2 the user can even configure customer data. When it comes to integration, Vendor 1 offers a bunch of extensions like:

- Browser Response Time Metrics
- CA APM for SOA
- ChangeDetector
- Environment Performance Agent
- IBM CICS Transaction Gateway
- IBM WebSphere Application Server Distributed Environments
- IBM WebSphere Application Server for z/OS
- IBM WebSphere MQ
- IBM WebSphere Portal
- IBM z/OS
- Microsoft SharePoint
- Oracle Databases

- Oracle WebLogic Portal
- Oracle WebLogic Server
- SiteMinder SNMP Collector
- SiteMinder Application Server Agents
- Web Servers

The Vendor 1 APM license allows the user to install all these additional extensions that increase the functionality of the solution. Vendor 2 offers monitoring, alerting and cloud auto-scaling extensions. The user can add more extensions, but, if there are no existing ones, they have to be programmed by your own. If the user needs help in creating a new custom extension, Vendor 2 offers you support.

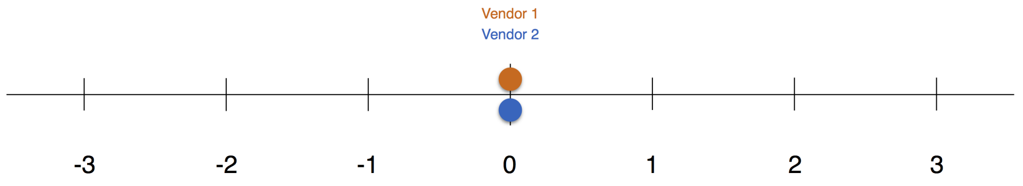


Figure 4.3. Assessment of use case no. 3

4.2.4 APM-UC-04 - Status-Quo Identification

Vendor 1

After initializing the agents as well as restarting the JVM, first results are available. Therefore, any frontend and backend are displayed automatically. Beside this step, the tools are displaying any collected information, metrics and errors out of the box. After receiving all the informations, the Team Center visualizes the top level overview of all components. Baselines and trends will be calculated after a while.

Vendor 2

As the agent starts reporting just after the installation is complete, first results can be seen just a few minutes later in the controller. In the case of System A , some slow calls could be found about 15 minutes after monitoring had started. Features like baselines and trends are not available that quick, the first baseline was available after about 1 day. However, it has to be mentioned, that these early baselines are not quite representative as there is just not enough data to calculate a decent baseline. Another feature which is available just seconds after the agent started reporting, is the *Flow Map*. It provides a visual representation of all components and also activities in the monitored application. Summing up, all (statistic) data is available immediately (respective a small delay due to data transport), only the analytic features like baselines and trends need some time to be calculated by the controller and much more important to be representative.

Conclusion

Important in this use case was to verify how fast the first results were present, how meaningful and extended the information are and how they are represented. Both environments are nearly on the same point. Both of them begin directly to collect useful data after the installation. In things of baselines, both vendors could actually make this feature available promptly, however it must be said that early results are not that significant. More important is to mention that both tools indicate errors and stalls right after the agent started reporting. Both tools display a rough outline of all components of the application quickly. In this use case both vendors are on the same point.

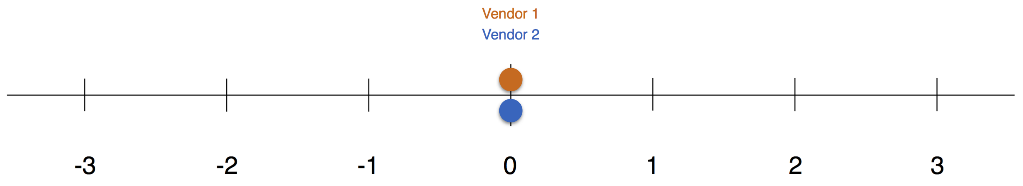


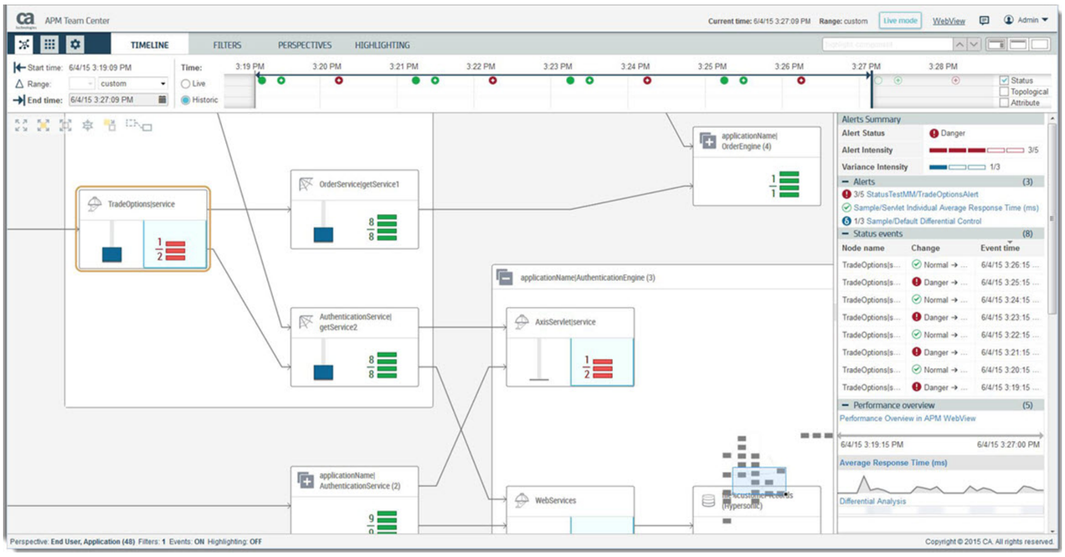
Figure 4.4. Assessment of use case no. 4

4.2.5 APM-UC-05 - Runtime Application Architecture

Vendor 1

Vendor 1 fulfills application discovery automatically. Therefore, the whole architecture, including all different components, is displayed in the Team Center tool. Using the tool will be the first step to initiate the Triage and the following drill down. This map gives information about:

- general health status of the system
- top-level architecture range
- dependencies between all subcomponents
- Trends
- Baselines



Current time: 6/4/15 3:27:09 PM Range: custom

Start time: 6/4/15 3:19:09 PM
Range: custom
End time: 6/4/15 3:27:09 PM
Time: ○ Live ● Historic

☑ Status
□ Topological
□ Attribute

Alerts Summary

Alert Status: Danger
Alert Intensity: 3/5
Variance Intensity: 1/3

Alerts (3)

- 3/5 StatusTestMM/TradeOptionsAlert
- Sample/Servlet Individual Average Response Time (ms)
- 1/3 Sample/Default Differential Control

Status events (8)

Node name	Change	Event time
TradeOptions|s...	Normal → ...	6/4/15 3:26:15 ...
TradeOptions|s...	Danger → ...	6/4/15 3:25:15 ...
TradeOptions|s...	Normal → ...	6/4/15 3:24:15 ...
TradeOptions|s...	Danger → ...	6/4/15 3:23:15 ...
TradeOptions|s...	Normal → ...	6/4/15 3:22:15 ...
TradeOptions|s...	Danger → ...	6/4/15 3:21:15 ...
TradeOptions|s...	Normal → ...	6/4/15 3:20:15 ...
TradeOptions|s...	Danger → ...	6/4/15 3:19:15 ...

Performance overview (5)

Performance Overview in APM WebView

6/4/15 3:19:15 PM 6/4/15 3:27:00 PM

Perspective: End User, Application (48) Filters: 1 Events: ON Highlighting: OFF

Copyright © 2015 CA. All rights reserved.

Figure 4.5. APM Team Center

To realize any sort of distributed monitoring, Vendor 1 provides the APM Command Center. The main function is the provision of an overview of CA APM agents in the respective environment. In general it gives the opportunity to:

- get a list of Java agents installed in the environment

- download agent log files
- obtain diagnostic information from a Java agent
- change the log level of an agent
- access its key resources via REST API

Relating to the use of many agents Vendor 1 uses the so-called Agent Controller. This Agent Controller acts as a point of presence on a server. Commonly, such an Agent Controller can manage multiple Java agents. Every agents attempts to connect to a socket opened by the Agent Controller itself. There is no connection between the different agents. Furthermore, agents do not save any information on their own.

Vendor 2

Vendor 2 provides fully automated architecture discovery without any effort needed for configuration. The central feature is the Flow Map. Components shown in a flow map are:

- Tiers (group of nodes),
- Nodes (e.g., a server),
- Message Queues and
- Databases.

Different line visualizations indicate different connection types between components, as there are:

- Synchronous connections and
- Asynchronous connections

Flow maps can appear in different contexts. The most important are:

- Flow Map showing all applications,
- Flow Map showing the topology of one application,
- Flow Map of a single tier or node,
- Flow Map showing information of one business transaction, and
- Flow Map showing metrics associated with a single snapshot.

This provides easy and quick access to snapshots and drill downs in order to see performamce problems within a few clicks. Custom flow maps can be created either by creating a new one or modifying an existing one, so that, e.g., a developer can create his own flow map showing only information relevant for his purposes.

Conclusion

Both vendors fulfill application discovery automatically and use a component graph to visualize the whole application architecture. Vendor 1 uses the APM Team Center to display the Component Map. Their map offers an overview of the dependencies and connections between all components of the application. Every component shows their actual status with baseline and alert. Vendor 2 visualizes the components and their dependencies with a Flow Map. The Flow Map distinguishes the types of the components and displays them in a way one can easily understand. In addition to that, the Flow Map differentiates between synchronous connections an asynchronous, a feature Vendor 1 does not support. Another benefit of the Flow Map is, that one can easily directly access snapshots and drill-downs by clicking on the component displaying an error.

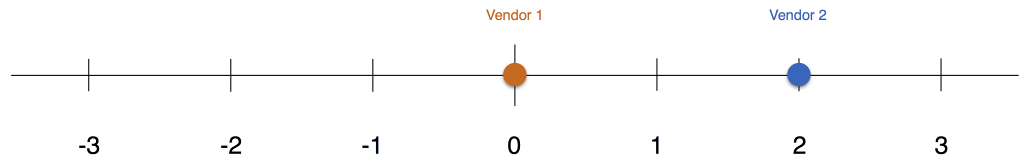


Figure 4.6. Assessment of use case no. 5

4.2.6 APM-UC-06 - Visualization / Dashboards

Vendor 1

The visualization of data is a very important fact to gather information quickly. Therefore, Vendor 1 gives backend as well as frontend overviews of the monitored system. To rehash the data, the tool can visualize the information in a Data Viewer by using different types:

- Graph
- Bar chart
- Graphic equalizer
- Dial meter
- String viewer
- Text viewer
- Alert

Vendor 1 also offers the use of dashboards. Therefore, pre-configured dashboards are available on the WebView and the Workstation. In this case the Web View Console provides views by using the dashboards that are available from the Enterprise Manager which WebView is integrated into. Those dashboards are not editable since WebView is a view-only client. To edit or create dashboards, the Workstation is necessary. This implies that Introscope Workstation provides to edit business logic constructs and view Metric data as well. It is possible to bookmark console dashboards for quick access. The sharing of dashboards is not supported. To get access to the dashboards, a ready-only user needs to be created. In additon, the Viewer Detail dialog can display the following details about each of the metrics displayed in the data viewer:

- Full metric name—The fully qualified metric name
- Min—The lowest value recorded for that metric
- Average—The average value for that metric
- Max—The highest value recorded for that metric
- Last—The most recent value recorded for that metric.

As another feature, Vendor 1 povides the visualization of their differential analysis approach to display information concerning the intensity, trends and alerting **??**.

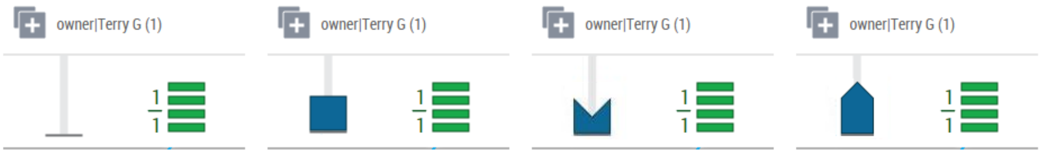


Figure 4.7. Differential Analysis

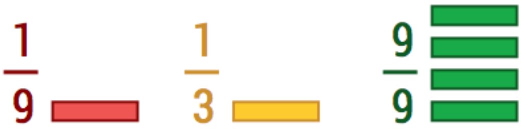


Figure 4.8. Alerting

- Green - none of the metrics for which the alert matches are breaching caution or danger thresholds
- Yellow - caution, an attempt to use the application or component is likely to yield unsatisfactory results
- Red - danger, a problem that requires immediate attention.

Vendor 2

Considering the wide range of functionality offered by Vendor 2, it is quite obvious that it takes some time to get familiar with the tool and the controller's UI. However, it has to be mentioned, that the process of learning Vendor 2 does not take ages. There are several reasons for that:

- For nearly every feature there is more than one way to access it,
- For beginners, there are many notes on how to work with Vendor 2 and
- The design is very intuitive.

Accordingly, the usability in terms of learnability is very good. When using the controller the first time, some tips and tricks are shown as pop-ups, facilitating the first steps.

Dasboards are an essential feature in general. Vendor 2 offers a wide range of dashboards. As usual the user can use one of the four predefined dashboards (e.g., the *DevDashboard*), or create custom ones. To get an impression of how dashboards look like

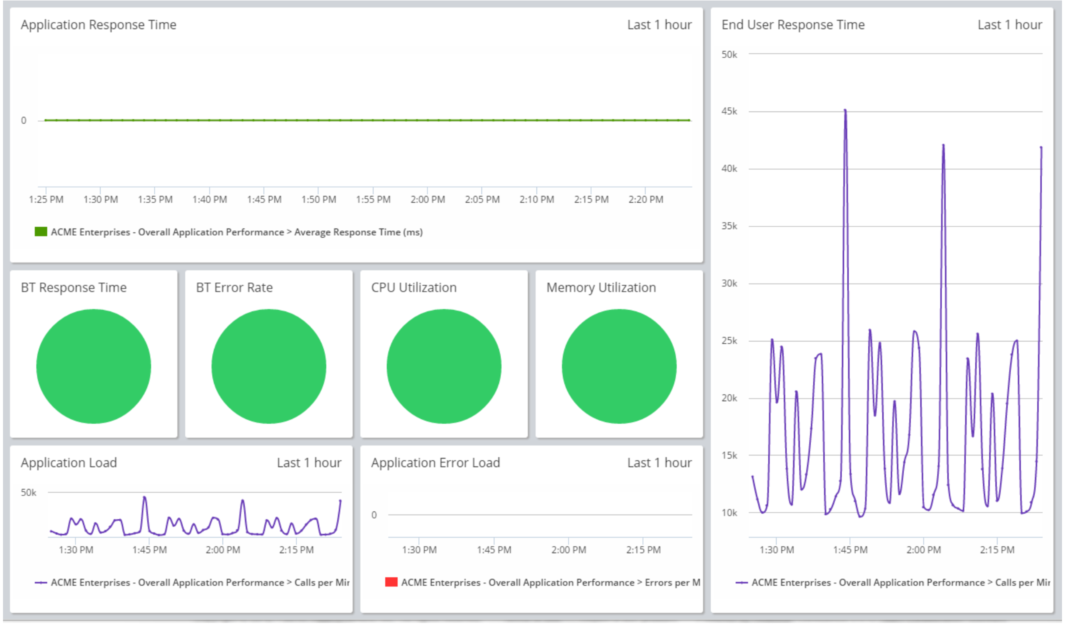


Figure 4.9. OpsManagement Dashboard

in Vendor 2's tool, Figure 4.9 shows a custom OpsManagement dashboard. Custom dashboards are built by drag-and-dropping the widgets needed into the dashboard space. Available widgets are:

- For metric data: graph, pie, gauge and metric value,
- For health rule and events: status light and list, and
- For Other data: label, IFrame and image.

For each widget you can define various properties and settings. The most important and useful are:

- Doubleclick action,
- Drilldown URL,
- Time range and
- Event overlay.

Dashboards can be exported and imported. Also the user can associate a dashboard from another node.

Vendor 2 offers options to customize the accessibility of dashboards. The user can therefore configure the (groups of) users allowed to view / edit / delete a specific dashboard. Differentiation between different user groups (e.g. first level support, power user and developer) is not provided. It his possible to implement such functionality. However, this takes a lot of configuration work.

Conclusion

First of all we have to mention that to create your own dashboards with the Vendor 1 solution you need the Vendor 1 Workstation. In addition to that we have to bring up that the Workstation didn't work at none of our threes' computer. (Windows and Mac User) Creating an own dashboard could take a lot of time with Vendor 1's APM. The Tool for designing is not very intuitive. Many alert icons and buttons must be dragged one by one on the edit view. After that you have to connect predefined alarms/alerts with the added items of the dashboard.

When it comes to creating dashboards, Vendor 2 offers the user a much better experience. Adding and positioning items is a lot faster thanks to the intuitive system. Just by clicking on a desired component, the solution automatically adds the item to a free space of the dashboard. Of course one can still edit the position an the size of the component manually.

In addition, dashboards can be exported and imported in the Vendor 2 solution.

In terms of visualization and dashboards Vendor 2 clearly got the point.

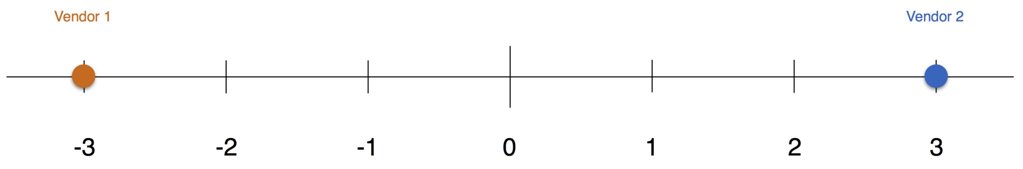


Figure 4.10. Assessment of use case no. 6

4.2.7 APM-UC-07 - Agent-Configuration

Vendor 1

In regard to agent configuration, Vendor 1 pursueds a three-part approach.

- standard configuration
- smart configuration
- individual configuration

These approaches allow the costumer to get a standard configurated agent without doing any changes or an agent where partially changes can be done. If there is need of individuality an individual configuration can be ordered. In general, configurations can be changed during runtime. Vendor 1 uses bpd. files as config files. Out-of-the-box the agent is optimized for the respective application environment. In addition, to some changes of configuration, the application needs to be restarted. Furthermore, all agents are reusable. Besides, Vendor 1 supports the surveillance of all agents to get an overall health care state.

Vendor 2

The configuration of Vendor 2's agents is done

- before downloading an agent, in the web interface of the controller , or
- by manually editing the config files of the agent.

If some configuration is done in the agents configuration file, the controller won't be able to override this configuration. That way you can, e.g., filter out sensitive data collected by the agent like passwords or also credit card numbers.

Vendor 2 does not offer preconfigured agent configurations optimized for some, e.g., specific web server like WebSphere. However, you do not need these preconfigured solutions because of the fact, that Vendor 2's agents are designed in a generic way and can be configured and optimized in the controller. The controller provides some features simplifying the agent configuration like, e.g., metrics specifically designed for JMX or special rules for JBoss or ActiveMQ etc. Moreover, the status of the agent is displayed in the controller. The agent can also be restarted or stopped if a problem occurs, e.g., if the application's performance decreases too much as a consequence of the instrumentation.

Conclusion

This use case was created to verify the possibilities of the agents' configuration. Vendor 1 as well as Vendor 2 are providing a manual configuration and an automatic configuration. The automatic configuration is not recommended for huge environments like the instrumented ones for this study. In the Vendor 1 solution the user can edit config files (.pbd files) to change the agent configuration. In the Vendor 2 solutiob it is easier to edit the configuration. Users do it right from the controller screen. In both solution all agents are reusable and you have a view of the health of the agents and users can restart them if problems are occurring. The instrumented application must be restarted to adopt the changes.

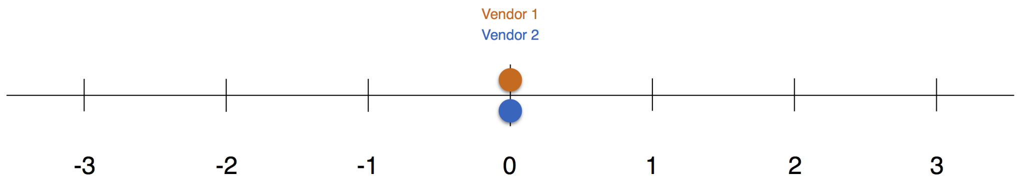


Figure 4.11. Assessment of use case no. 7

4.2.8 APM-UC-08 - Hardware Monitoring

Vendor 1

In regard to the resource management and monitored hardware, Vendor 1 provides a lot of metrics out of the box. Templates for overall system information are appropriate. In general, the tool reports the 5 important metrics:

- Average Response Time
- Concurrent Invocations
- Errors Per Interval
- Responses Per Interval
- Stall Count

Among others, the tool can display metrics sorted in the categories below:

- Memory-Related Metrics
  - GC Monitor metrics
  - GC Heap metrics
- Utilization metrics
  - CPU:Utilization (process)
  - CPU:Utilization (aggregate)
- Socket metrics
  - Current number of deadlocked threads
- Thread Dump Metrics
  - I/O threads
  - Worker threads
- Connection pool metrics
- Event metrics
- Resource Metrics
- Agent Stats

Vendor 2

In order to monitor hardware, Vendor 2 provides machine agents. Among others, the following metrics are available:

- Availabiliy,
- CPU,
- Memory,
- Network and
- Disk.

These metrics all have a bunch of sub-metrics, e.g., the metric *CPU* has a sub-metric *Busy*. In addition to that,

- Processes

can be monitored likewise. Information about which processes are active, how much memory they are using, the PID, the status etc. are available.

JVM monitoring is also possible. Important metrics are among others:

- Heap (size, usage),
- Garbage-Collection,
- Process burn,
- Connection pools and
- Thread pools.

Conclusion

In things of hardware monitoring both environments provide the user with a bunch of metrics. Metrics like CPU, memory, availability and a lot of sub-metics for processes were offered by Vendor 2 and Vendor 1 . Both of them support JVM Monitoring. To actually view all these metrics in the Vendor 1 environment, you need to open up the WebView.

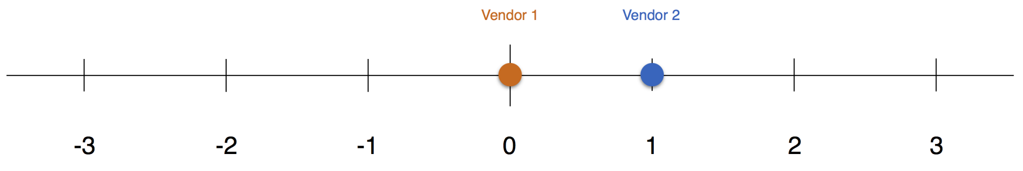


Figure 4.12. Assessment of use case no. 8

4.2.9 APM-UC-09 - Root Cause Analysis / Deep Diving

Vendor 1

For root cause analysis and the associated deep diving, Vendor 1 supports the generation of thread dumps. Information about the thread state are visualized by graphs. The tool displays a list of waiting, hanging or running threads. Tracing can be done manually by using the trace view. This view gives information about the length of SQL statements and information about methods and classes. When errors occur, an error snapshot can be created by using the stackview whichs gives a detailed view about:

- Trace ID
- Class
- Exception

Eventually, all errors will be displayed whereby recognition of already occured errors is not provided. Concerning the identification of root causes, filter possiblilities are supporting the retrieval. The following filters are available:

- Time
- Type of error
- BT query

For generating automatization, manually configured shell scripts can be used.

Error treament has to be configured manually. To accomplish any sort of triage, error searching as to be done manually by using dashboards or topological maps. To conduct any business transaction analysis, the WebView and TeamCenter are at disposal. Occured alerts can be appreciated in the TeamCenter. Therefore, the tool uses a timeline to display errors or general state changes (status event) in different colors. Moreover, the selection of several timeframes is available. Based on this timeline, problems can be detected easily. So that the actual drill-down can be performed, the WebView is necessary to check the respective components by using the displayed metrics. Considering the metrics, the agents collect up to 3000 different metrics. Referring to any client monitoring, Vendor 1 supports:

- Mobile App Analytics(MAA)

  ⇒ agent runs on the device

- Browser Monitoring

  ⇒ Java Script Snippets are used

- FAT-Client

Vendor 2

The identificaton of, e.g.,

- slow methods,
- badly performing SQL-statements, and
- messages,

and in general performance problems happens within a few seconds — approximately five to ten seconds, acting on the assumption of good network latency. The performance of the monitored application is hardly affected, as the overhead will statically stay under 2%, no matter how high the current utilization is (see also section 4.2.14).

All kind of metrics are available, listing them would go beyond the scope of this work. More interesting is the fact, that the user can create his own metrics by declaring an analytic search query as a new metric, which can be monitored in the same way as every other predefined metric.

Another very useful technique are filters. They can be set for nearly every result set shown in the controller. In addition to some predefined filters (e.g., type: POJO, Servlet, JMS, etc.), Vendor 2 allows filtering based on plain text input (e.g., regular expressions).

Vendor 2 defines a business transaction as an "end-to-end processing path used to fulfill a service request in the monitored environment"[1]. Business transactions are automatically created based on a request at an entry point in the application, which is tracked through all involved components of the application environment.

The user can analyze business transactions more deeply by comparing two of them side by side. That makes sense, if there is, e.g., a method performing differently in several cases. Similar to that, there is a feature called *Relevant Fields* allowing users to compare fields like, e.g., the average response time after a filter has been applied to a set of business transactions.

Drill downs of transaction snapshots, as already mentioned, can be accessed with only a few clicks in several ways. Besides the actual call graph, there is additional information, like, e.g.:

- Potential issues (a list of the most critical parts of the business transaction),
- A business transaction flow map (s. 4.2.5),
- SQL calls,
- HTTP data,
- Node problems (is current data drifting away from a baseline),
- Error details (e.g., the file name causing an error), and

[1] https://docs.appdynamics.com/display/PRO41/Business+Transactions

- Hardware / memory information.

Information points allow the collection of data based on a specific point in code across all business transactions. This is useful when it comes to the correlation of application performance and business performance (e.g., how many credit cards did System B process over a fixed time range or what was the average time spent for configurating a new car with System B).

At this point, it has to be mentioned, that client monitoring is done the same way. Database monitoring provides some additional features, the most important are:

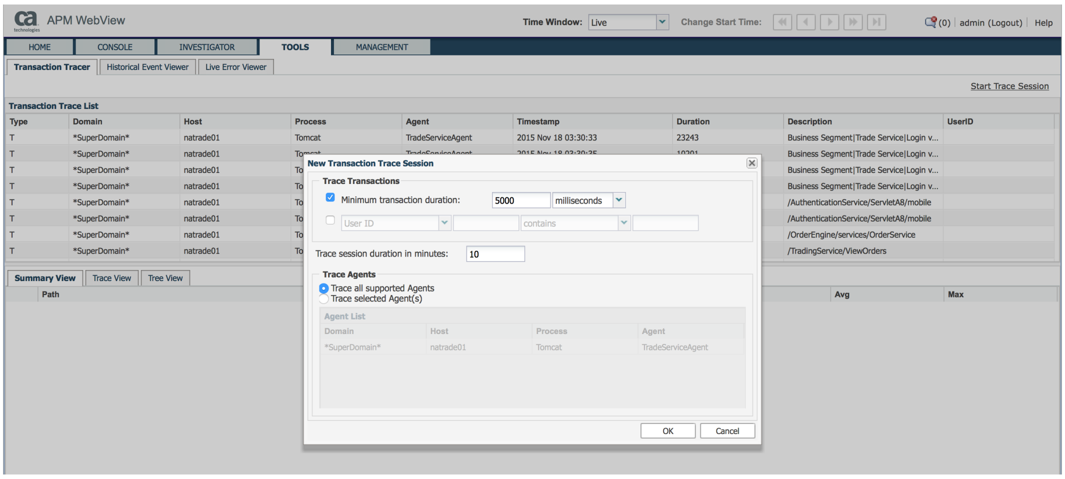
- Database dashboards displaying detailed information for each individual database,
- Live-View (shows the current activity for a database server - CPU / Mem Usage, SQL wait states, session list),
- Queries window (shows the SQL statement consuming the most CPU time in the database), and
- Query execution plan window (can be helpful determining the most efficient execution plan for a query).

All metrics and baselines are also available for databases. Alarms (see also section 4.2.11) can be set equally.

Vendor 2 allows the analysis of log files (unstructured data). This feature is not ready out-of-the-box. Before being able to capture log data, a log source has to be configured for the analytics agent. Once this is done, the user can convert this unstructured data into new metrics. Furthermore, the user can define new filter criteria based on log data — e.g., the log level. Log data can also be accessed from the call drill down of a business transaction — it simply creates a new analytic search with a preconfigured filter.

Conclusion

"Where is the error?" and "How can I find the Root-Causes?" were the main questions in this use case. Both applications are able to identify slow methods, hanging threads, not performant SQL-Statements & response time. Both of them allow the user to trace & drill-down any occurred transaction. In the end of the day we would recognize that Vendor 1 as well as Vendor 2 support all the listed criteria. The interesting part of this use case is in the question "How can I find the root causes?". Let us begin with Vendor 1 : To catch a transaction the user needs to open up the WebView. The WebView is offering a tool named "Transaction Tracer" 4.13 where you can start a trace session. Before the application starts tracing, the user is allowed to enter some specific filter (e.g. transaction duration, trace agents, etc.) to reduce the amount of transactions.



Transaction Trace List

Type	Domain	Host	Process	Agent	Timestamp	Duration	Description	UserID
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:30:33	23243	Business Segment|Trade Service|Login v...	
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:30:35	10201	Business Segment|Trade Service|Login v...	
T	*SuperDomain*	natrade01	To				Business Segment|Trade Service|Login v...	
T	*SuperDomain*	natrade01	To				Business Segment|Trade Service|Login v...	
T	*SuperDomain*	natrade01	To				/AuthenticationService/ServletA8/mobile	
T	*SuperDomain*	natrade01	To				/AuthenticationService/ServletA8/mobile	
T	*SuperDomain*	natrade01	To				/OrderEngine/services/OrderService	
T	*SuperDomain*	natrade01	To				/TradingService/ViewOrders	

Path	Avg	Max

New Transaction Trace Session

Trace Transactions

☑ Minimum transaction duration: 5000 milliseconds

☐ User ID contains

Trace session duration in minutes: 10

Trace Agents

● Trace all supported Agents
○ Trace selected Agent(s)

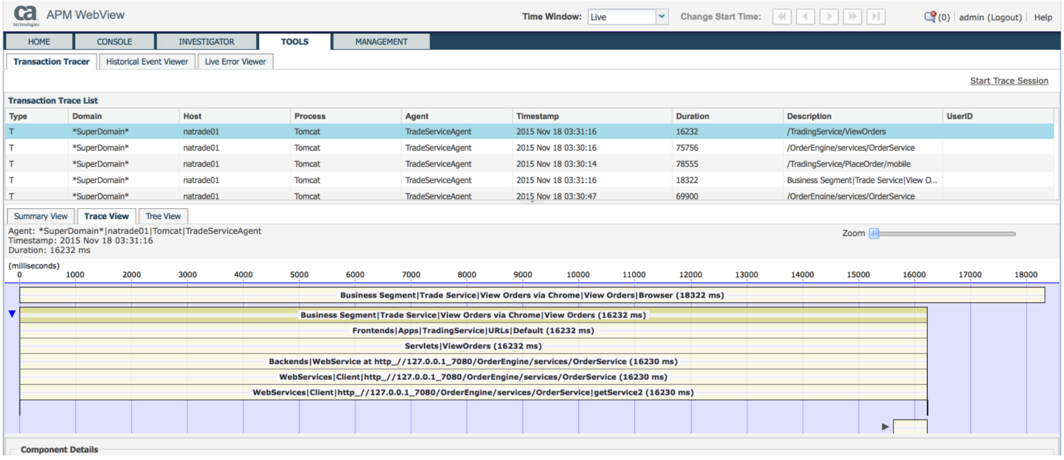
Agent List

Domain	Host	Process	Agent
SuperDomain	natrade01	Tomcat	TradeServiceAgent

OK Cancel

Figure 4.13. Vendor 1 Transaction Tracer

After setting the filter and starting the trace session, the Application starts to find the specific transaction traces and displays them in a list, where the user can choose the wanted one to gather more particular information 4.14 (e.g. Trace View).


APM WebView
Time Window: Live
Change Start Time:
(0) | admin (Logout) | Help
HOME
CONSOLE
INVESTIGATOR
TOOLS
MANAGEMENT
Transaction Tracer
Historical Event Viewer
Live Error Viewer
Start Trace Session
Summary View
Trace View
Tree View
Zoom
(milliseconds)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
Business Segment|Trade Service|View Orders via Chrome|View Orders|Browser (18322 ms)
Business Segment|Trade Service|View Orders via Chrome|View Orders (16232 ms)
Frontends|Apps|TradingService|URLs|Default (16232 ms)
Servlets|ViewOrders (16232 ms)
Backends|WebService at http_//127.0.0.1_7080/OrderEngine/services/OrderService (16230 ms)
WebServices|Client|http_//127.0.0.1_7080/OrderEngine/services/OrderService (16230 ms)
WebServices|Client|http_//127.0.0.1_7080/OrderEngine/services/OrderService|getService2 (16230 ms)
Component Details

Transaction Trace List

Type	Domain	Host	Process	Agent	Timestamp	Duration	Description	UserID
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:31:16	16232	/TradingService/ViewOrders	
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:30:16	75756	/OrderEngine/services/OrderService	
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:30:14	78555	/TradingService/PlaceOrder/mobile	
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:31:16	18322	Business Segment|Trade Service|View O...	
T	*SuperDomain*	natrade01	Tomcat	TradeServiceAgent	2015 Nov 18 03:30:47	69900	/OrderEngine/services/OrderService	

Agent: *SuperDomain*|natrade01|Tomcat|TradeServiceAgent
Timestamp: 2015 Nov 18 03:31:16
Duration: 16232 ms

Figure 4.14. Vendor 1 Trace View

This graph shows the duration of the chosen transaction and explains how much time every component spent, so that we could distinguish healthy and unhealthy components. Furthermore, the Trace View displays even method calls and the corresponding class.

In addition to that, the APM solution of Vendor 1 offers alert views to every system component right in the Team Center view.
Another feature is the Vendor 1 APM Timeline **??**, where the user easily can notice changes of alert status, topology of the system and attributes. With this feature, the user can recognize when an error occurred and if it was after an update of the system or after a topology change. This feature does not exist in the Vendor 2 solution.

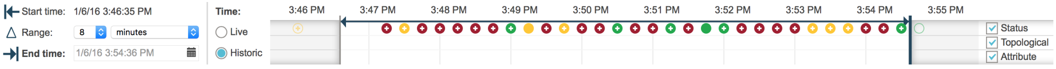
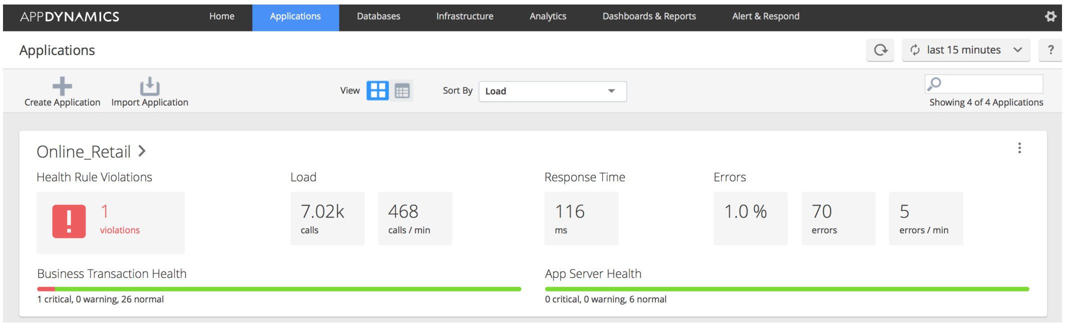

Start time: 1/6/16 3:46:35 PM
Range: 8 minutes
End time: 1/6/16 3:54:36 PM
Time: ○ Live ● Historic
☑ Status
☑ Topological
☑ Attribute

Figure 4.15. Vendor 1 Timeline

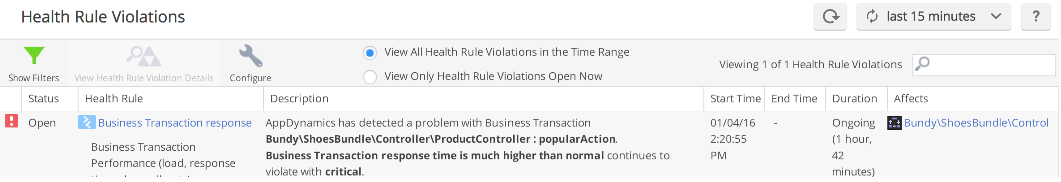
With Vendor 2, finding the cause of an error is an easy thing. The user has many possibilities in doing it. The first option is right on the Application screen **??**. There is a list of all monitored applications with their health status. If we have a problem in an application, the health status will be red. Assumed, that we have an error as in the example. The only thing the user needs to do is to click on the health status.


Applications
last 15 minutes
View
Sort By Load
Showing 4 of 4 Applications

Online_Retail
Health Rule Violations: 1 violations
Load: 7.02k calls; 468 calls / min
Response Time: 116 ms
Errors: 1.0 %; 70 errors; 5 errors / min
Business Transaction Health: 1 critical, 0 warning, 26 normal
App Server Health: 0 critical, 0 warning, 6 normal

Figure 4.16. Vendor 2 Application Screen

After clicking the health status **??** in the application view, the tool will send the user right to a list of all detected errors with information like status, description, time and duration and what this error is affecting.



Health Rule Violations

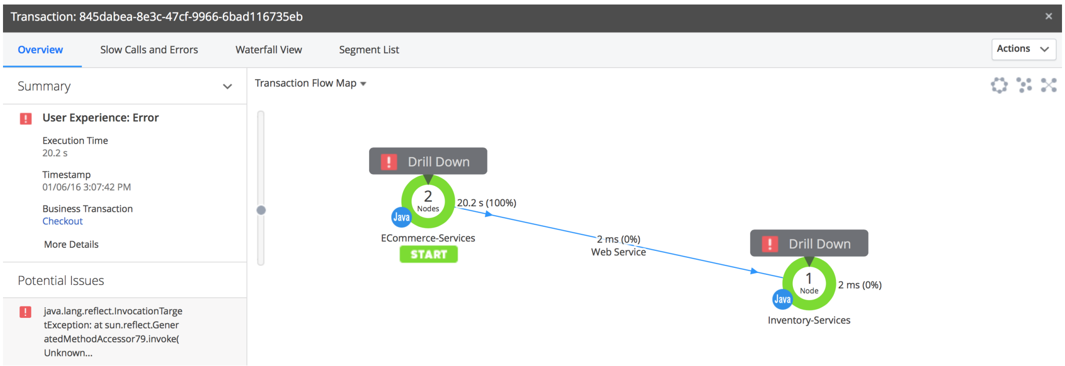
● View All Health Rule Violations in the Time Range
○ View Only Health Rule Violations Open Now

Viewing 1 of 1 Health Rule Violations

	Status	Health Rule	Description	Start Time	End Time	Duration	Affects
!	Open	Business Transaction response Business Transaction Performance (load, response	AppDynamics has detected a problem with Business Transaction **Bundy\ShoesBundle\Controller\ProductController : popularAction**. **Business Transaction response time is much higher than normal** continues to violate with **critical**.	01/04/16 2:20:55 PM	-	Ongoing (1 hour, 42 minutes)	Bundy\ShoesBundle\Control

Figure 4.17. Vendor 2 Health Rule Violation List

Another option is to look for slow calls. Therefore, the user opens the menu Transaction Snapshots. A list with all snapshots comes up. We have a possibility to filter transactions. Now the user can simply chose a slow transaction and view the snapshot. The user gets a screen with information about the selected transaction and even indicates potential issues. The user are even able to drill-down from this view.



Transaction: 845dabea-8e3c-47cf-9966-6bad116735eb

User Experience: Error

Execution Time
20.2 s

Timestamp
01/06/16 3:07:42 PM

Business Transaction
Checkout

More Details

Potential Issues

java.lang.reflect.InvocationTargetException: at sun.reflect.GeneratedMethodAccessor79.invoke(Unknown...

Figure 4.18. Vendor 2 Snapshop View

The drill-down report can be exported as a PDF File.

In conclusion we have to mention that these strategies of finding the root cause of an error are not all and that there are other ways of finding it. Both vendors cover all the criteria listed in the definition of the use case. The intention of showing the root cause analysis strategies of both solutions side by side was to make clear which one of the application is easier and more intuitive to use. In terms of usability, Vendor 2 gets the point.

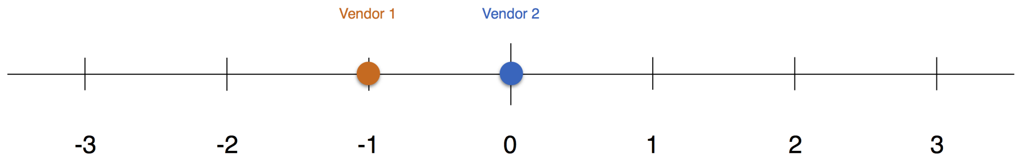


Figure 4.19. Assessment of use case no. 9

4.2.10 APM-UC-10 - Analytics

Vendor 1

To analyze the collected data, Vendor 1 uses automatically as well as manually configured baselines. Any trends are recognized automatically by using collected metrics and information during the past. One important feature is the differential analysis. The differential analysis is an approach for automatically identifying important changes in the performance of an application. To fulfill this analysis it is necessary to search for uncontrolled variance by using the average response time and business transaction metrics. At least a trend can thus be identified. The tool offers the possibility to get the trends and baselines in different time periods manually or automatically. By using the timeline in the Team Center, the start and end of a time period can be adjusted. Another possibility are report templates. With these reports it is posible to get trends:

- Daily
- Weekly
- Monthly
- or in any time periode

The WebView tool provides the export of a snapshot of current data in a Data Viewer to a comma-separated values (.csv) file as well as the PDF export of reports. In general, the report does not contain new events like application changes. Those changes can be detected by using the metric count. If an event occured, the number of metric count will be changed. However, the timeline visualizes new occured events as topological events. It gives informations about:

- Node name
- Change
    - Node added
    - Node deleted
    - Connection added
    - Connection deleted

Vendor 2

Baselines are a feature, which probably is under the top five of the most important and valuable ones. Vendor 2 supports baselining with two different base time periods:

- Fixed time range, and

- Rolling time window (e.g., always the last x days).

The second variable of a baseline is the hourly segment based on which it is (continuously) calculated and updated. It can be either unsegmented or segmented. In the latter case, the following trends are available:

- Daily (e.g. everyday 10am - 11am),
- Weekly (e.g. every monday 10am - 11am), and
- Monthly (e.g. every first of a month 10am - 11am).

Basically, baselines can be configured for every metric available. Custom baselines can be created likewise.

Vendor 2 manages the monitoring of application clusters by grouping several nodes into tiers. In an analytic search, you can create a view for only a subset of all nodes. These can be chosen from all nodes in all applications.

Following is a list of things that can be exported:

- Current configurations as XML,
- Metrics as CSV / PDF, and
- Reports as PDF.

The import and export of the items in the above list can be automated by using the provided REST API. Also the user can export transaction snapshots, but it has to be mentioned, that Safari crashed multiple times during the export of a transaction snapshot.

Another feature that allows fast troubleshooting, is an event. Events can be:

- Health rule violations,
- Errors,
- Code problems (e.g., a deadlock),
- Application changes (e.g., deployment, app server restart), and
- Application discovery (e.g., new backend, new business transaction).

Detailed information about every event is accessible, e.g,. in an app server dashboard. In addition to that, a flow map for only the specified event is available.

Conclusion

In this use case, the main point was to research the possibility of analyzing the collected data. Keywords like baselines, trends and import and export functions were the main criteria for this use case. It is clear that the algorithms of creating a caseline and trend are patented in both cases, therefore we can only give information of what is affecting a baseline or a trend. Vendor 1 use for "baselining" the feature differential analysis. In addition, both vendors are able to recognize trends automatically. These recognized trends are affecting the baseline within past collected data. Both APM solutions save historic data to predict, with some deviations, what will happen in near future. Vendor 1 as well as Vendor 2 allows to evaluate nearly every diagram (created from measured metrics), to export them to CSV and to export reports in PDF. Vendor 2 can export current configurations as XML, Vendor 1 ca not. In addition, Vendor 2 offers an automated import and export of the described items, by using a REST API, Vendor 1 does not support this feature. In conclusion we have to say that both nearly are on the same point, when it comes to analytics, but Vendor 2 offers a few more features, that Vendor 1 does not.

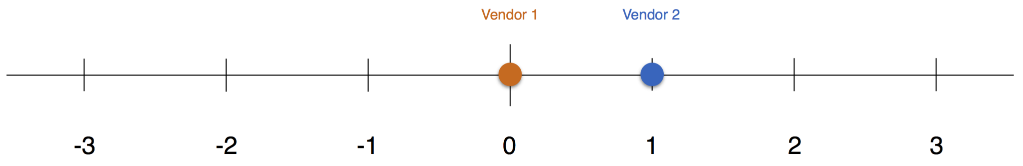


Figure 4.20. Assessment of use case no. 10

4.2.11 APM-UC-11 - Pro-Active Monitoring

Vendor 1

To realize pro-active monitoring, the Vendor 1 tool allows setting thresholds on every sort of metrics. Therefore thresholds are also divided in two categories:

- danger value
- caution value

Those values are manually adjustable. It is also possible to set a comparison of attributes by using the equal option. In order to respond to posssible threshold exceedance, alerts can be set. By using summary alerts, different alerts can be merged as one. Relating to those alerts and associated automatization, the tool offers actions like automatic email, notifications. For the summary alarm, different trigger options are available. Thus the alert can be triggered if selected subalerts are triggered.

Vendor 2

Thresholds can, among others, be set for

- every metric,
- every business transaction and
- every action

Moreover, the user can define so-called soft thresholds. They can be configured for dashboards and only affect the visualization.

Vendor 2 comes with many preconfigured settings (thresholds), in order to inform the user, if a deviation of a baseline occurs, aiming to give the user time to prevent an error even before it actually occurs (e.g., users can get an email with a direct link to a dashboard providing all information needed). This is interesting, beacuse, if System A becomes to slow or data is lost on some transfer, this can result in a delay of production, again causing a sales decrease.

Health Rules are a mechanism to define the health of an entity like, e.g., a node or a CPU. This health status is calculated based on metrics like, e.g., the average response time. There are several health rules available:

- Transaction performance,
- Node health,
- End user experience & mobile user experience,
- Databases & remote services,

- Error rates,
- Information points, and
- Service endpoints.

Furthermore, the user can define custom health rules in order to evaluate custom metrics. Health rule schedules allow the user to manage the time schedule, when a health rule shall be active, which data it is based on and if there should be additional health rules active during a violation.

Conclusion

With this use case we wanted to research, how much the application helps the user. Points like automatic reaction, alarm management and alerting were important in this use case. Both vendors use thresholds in order to realize pro-active monitoring. It is possible, for both applications, to set a summary of different alerts that can appear if there is a transgression of a threshold. Vendor 2 can set thresholds on every metric, every business transaction and every action. They also offer soft thresholds, that can be configured for dashboards. In addition, Vendor 2 has an other feature named health rules, which can define the health of a node, CPU or transaction performance. A feature similar to the health rules does not exist in Vendor 1 's solution.

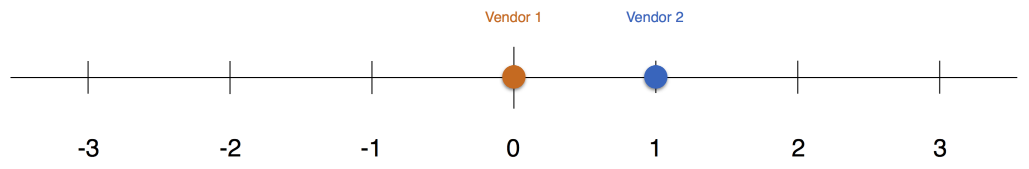


Figure 4.21. Assessment of use case no. 11

4.2.12 APM-UC-12 - Automation

Vendor 1

In general, automation is used to react to any occurred error. In reaction to those errors, Vendor 1 offers different actions:

- SNMP alert
- Skript actions
- Notification actions (email, console, pop-up window)

It is possible to create templates for automated workflows. Actions accomplished by any scripts just need to be created once. Sophisticated Reactions can be realized using the actions mentioned before. Therefore, certain know-how is recommended.

Vendor 2

Actions are the Vendor 2 method to react automatically to errors / problems. There are the following kind of actions:

- Notification actions (e.g., SMS, email, etc.),
- Diagnostic actions (e.g., a thread dump, diagnostic session).
- Remediation actions (e.g., a script / executable), and
- Custom actions (based on the first three action types custom actions can be defined).

With policies, there is a mechanism to automate reactions to monitoring results. On the one hand, you can define actions to be executed if, e.g., a health rule violation occurs and, on the other hand, also be able to suppress an action if a harmless error occurs frequently and can be ignored.

There is one more kind of action not mentioned in the above list, which is cloud auto scaling. Such actions allow the user, e.g., to dynamically start up / shut down additional servers or to role out images. Vendor 2 provides a so called task library, aiming to simplify and shorten frequently executed tasks, e.g., there is a task called addTomcatWorker starting a new tomcat instance.

Workflows represent a useful technique to bundle multiple tasks. All tasks available in the task library can be used in a workflow. Vendor 2 offers some preconfigured workflows as well as the possibility to create new ones.

All of these automation mechanisms can be applied to every instance being monitored.

In addition to all that, the REST API realizes automation in the context of integrating data into other IT-Systems an enterprise is running (e.g., reporting systems) (see also 4.2.3).

Conclusion

Important in this use case was to see, if the Vendors' tools were able to set automatic reactions or templates for automatic workflows. Both support notification and script actions. Vendor 2 's solution has the above mentioned feature cloud auto scaling. Furthermore, you can create workflows to bundle multiple tasks. Vendor 1 doesn't have a similar features like the mentioned ones in their solution.

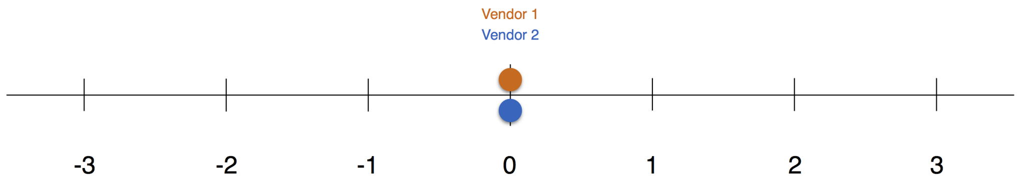


Figure 4.22. Assessment of use case no. 12

4.2.13 APM-UC-13 - Scalability

Vendor 1

In addition to the scalability, Vendor 1 uses two different approaches:

- Standalone Enterprise Manager
- Cluster

The enterprise manager can be used as a standalone component. In case of large application environments, Vendor 1 offers the opportunity to create a cluster by using many enterprise manager to collect the data from the agents. Those enterprise managers operate as a collector. Each collector receives the information from certain agents. It is recommended to use 300-500 agents per controller.

In general, the hardware requirements are:

- Windows, Linux, Solaris
- 2 Cores
- 4 GB RAM
- Memory depends on data size

Vendor 2

The number of agents per controller eventually depends on the metrics (used). Vendor 2 states 15.000 agents as a guiding value. Note that this value is not mandatory. In the case of more than 15.000 agents, the controller can be split into two or more instances appearing as only one controller.

Conclusion

In this use case we wanted to research the scalability of the tools. Both vendors are using different approaches in terms of scalability. Vendor 1 will allow the user to create more collectors if more agents are needed. Vendor 2 will allow the user to split the agent controller in more instances if more agents are necessary. It is clear that the strategy of both vendors is different, but the results are the same, both are offering a wide scalability.

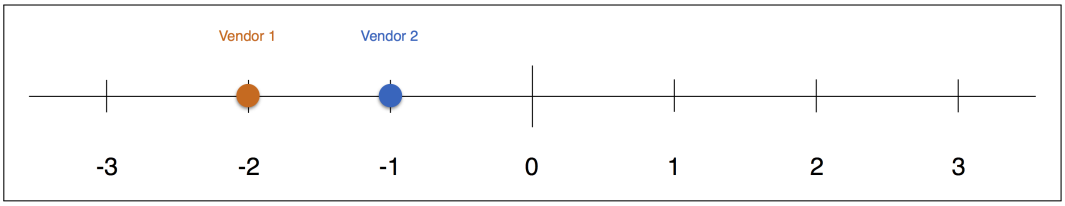


Figure 4.23. Assessment of use case no. 13

4.2.14 APM-UC-14 - Agent Functionality / Data Collection

Vendor 1

In regard to agent functionality and technical aspects, Vendor 1 uses 2 different strategies:

- Smart instrumentation
- Manual instrumentation (.pbd files)

In general, Vendor 1 uses tracers placed on the beginning and end of each method. Based on those tracers the path of a trace can be determined. The tracers can be placed manually or automatically via smart instrumentation. By using the smart instrumentation, the code with complex methods as well as the characteristics of each method will be recognized automatically. In regard to the produced overhead by the agents, an amount of up to 4-6% on CPU and memory is possible. But in general, it depends on the application.

Vendor 2

Vendor 2 agents collect a wide range of data, the following list provides an overview of the most important aspects:

- Metrics (also foreign metrics),
- Snapshots,
- Analytics (transactions and logs),
- Database,
- Broser and mobile real user monitoring, and
- Server data.

Vendor 2 says, that the overhead produced by their agents, is lower than 2%. In order to achieve this goal, a combination of both

1. bytecode injection and
2. call stack sampling

is used. Figure 4.24 shows the technique, Vendor 2 uses to reduce the overhead produced by the agent. The red fields are the points where instrumentation actually happens. Only the beginning and the end of, e.g., a longer running thread is instrumented, the rest is inspected directly on the callstack (by the agent). Compared to instrumenting "everything" (full instrumentation), this obviously reduces the agent overhead.

Furthermore, Vendor 2 allows some kind of manual instrumentation. The user can set / define

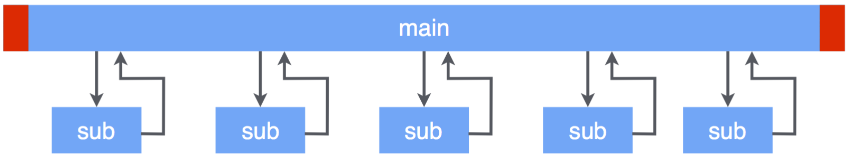


Figure 4.24. Instrumentation

- Information points and

- Data collectors

Information points are a way to manually define a point in code in order to collect data, starting from this very point accross all business transactions, whereas data collectors only collect data in the context of one business transaction.

Conclusion

Vendor 1 as well as Vendor 2 use bytecode instrumentation to collect data. Vendor 1 uses tracers and amart instrumentation. Tracers determine the application components to monitor. Tracers are defined in ProbeBuilder Directive PBD-Files. During the installation, some default PBD files are out-of-the-box to ensure immediate monitoring of the application. By editing the PBD files, one can modify, how the application is instrumented.
The java agent inserts probes into the bytecode of the instrumented application (bytecode instrumentation).
Smart instrumentation uses low overhead tracers that allow agents to discover and automatically instrument additional methods (without PBDs). This is possible, thanks to filtering and thanks to a (stack trace / call trace). Instead of full-instrumenting the application, smart instrumentation identifies important methods e.g. by their complexity and filters unimportant methods. The next point is that highly complex methods are pushed into a call stack were only the entry and exit time is measured. Introscope uses smart instrumentation to automatically collect transaction traces.
Vendor 2 uses a combination of bytecode injection and call stack sampling, so that the overhead produced by the agents can be maintained low. "'Vendor 2 runs in production with less than 2% overhead on the application. Vendor 1 with up to 4-6%.
As we can see both vendors nearly use the same strategies for instrumenting an application. With a lower overhead of 2%, we can assume that Vendor 2 ′ agents are working more efficiently, than Vendor 1 ′s agents.

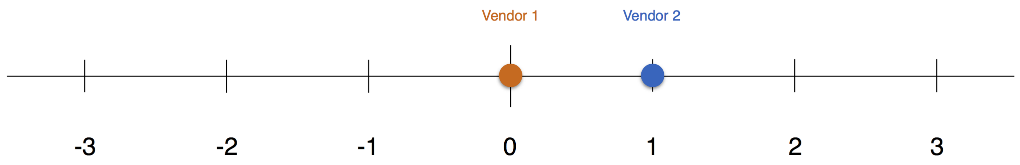


Figure 4.25. Assessment of use case no. 14

4.2.15 APM-UC-15 - Privacy

Vendor 1

In general, privacy had highest priority. Therefore, Vendor 1 uses encryption and certificates for certain transmitions. Transmitions with personal data will be masked. It is also possible to configurate the storing of data. Out-of-the-box, no sensitive data will be collected. For transmissions, Vendor 1 uses a proprietary protocol. It is also possible to change the transmission to HTTPS.

Vendor 2

Vendor 2 by trend is able to capture all data including passwords. Using the out-of-the-box configuration, no sensitive data is collected and recorded. If wanted, the collection of sensitive data has to be configured in the controller, special permissions are required. Additionally, Vendor 2 tracks all changes made to configurations. Sensitive data can be displayed with ***.

The transmission of the data collected by the agent and the controller interface can be done with HTTPS. Note that communication between server and controller is unidirectional.

Conclusion

In things of privacy, both vendors are on the same level. Vendor 1 as well as Vendor 2 are using encryption and certificates for data storage and transmission.

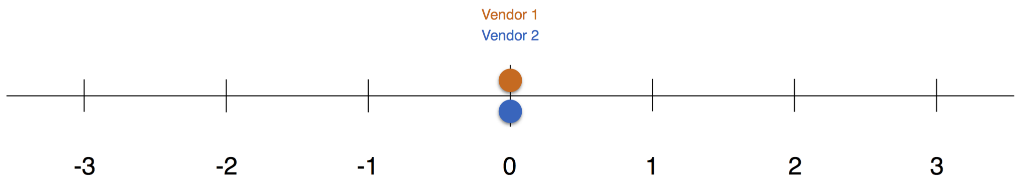


Figure 4.26. Assessment of use case no. 15

Chapter 5

Application Use-Cases

This section features some use cases related to the first of Gartner's 5 Dimensions of APM: End-User-Experience-Monitoring.

5.1 Use-Cases

The following list shows the five realistic scenarios, selected for evaluation:

- Application restart,
- Clickpath of a System B configuration,
- Buildability check in the context of System B ,
- PDF export, and
- CDB-invalidation.

5.2 Use-Case-Evaluation

5.2.1 APP-UC-01 - Application Restart

Vendor 1

In this scenario, the restart of the System B has to be recognised by using Vendor 1 tools. In regard to the occured event, the tool was not able to detect explicitly a restart. However it is possible to detect such an event. Therefore the connection state metric was used to recognize any changes. With the corresponding graph the restart could be identified. In general an alarm with a threshold can be set on the following metric to get automatically any notifications.

Vendor 2

The restart of the System B could not be discovered with the default (out-of-the-box) configuration. Therefore, it is necessary to set a flag in the javaagent parameter. Once this is done, Vendor 2 is able to monitor the application in terms of discovering application restarts. The controller did show a change event after the restart with information about, (among others):

- Tiers,
- Nodes,
- Changed properties and
- Startup options.

Allthough the check.jsp, used for checking the application status after a restart, did not show any errors, Vendor 2 agents discovered an error.

5.2.2 APP-UC-02 - Clickpath of a Configuration

Vendor 1

To get any informations about the clickpath of a user, Vendor 1 uses a new transaction trace session. After that, the transaction trace viewer will be consulted to get specific information. It can also be used to start any deep diving.

Vendor 2

Vendor 2 provides a wide range of information regarding the clickpath of an user. The session ID can be tracked, so the behaviour of users (customers) can be tracked and analyzed likewise. This is helpful, if, e.g., there is an error with some specific feature on a website leading customers to leave the site. With Vendor 2 , the user is able to find these critical points on a websites and to know, where to immediately fix some problems or optimize, e.g., a workflow.

After limitting the bandwidth to a very slow 5 KB/s, the decreasing performance was detected and shown in the controller UI within the standard delay of about 2 minutes.

5.2.3 APP-UC-03 - Buildability Check

Vendor 1

To analyze the buildability check Vendor 1 uses the transaction trace approach. Therefore, a new transaction trace session has to be made. The filter options helps to glean single transactions about the buildability check. To get more information the trace graph has to be used.

Vendor 2

The analysis of a buildability check with Vendor 2 provides the following information:

- There is an end-user-response-time of 4,8s.
- The actual buildability check is not the problem.
- The network was eliminated as a root cause.
- The cache and the database calls (only 6%) can be eliminated likewise.
- The hardware performed also quite well.
- The browser snapshot implicated, that most of the time is spent in the backend (3,7s). Therefore, this probably is the root cause.

A snapshot for the backend was available. To drill down, an agent in the backend is needed.

In order to monitor the CC-jar (important component in System B), a custom exit point had to be defined by manually looking for calls to the CC-jar in the transaction snapshots.

5.2.4 APP-UC-04 - PDF Export

Vendor 1

For monitoring the PDF export of car configurations, it is necesarry to filter the transaction again. Therefore, also the trace view was used. The Vendor 1 tool detected a servlet transaction with a duration of 4 seconds. During the monitoring, it was approved, that the database access was not the problem. The loss of time occured due to long backend calls. In general, the problemzone could be recognized.

Vendor 2

For monitoring the PDF export of car configurations, Vendor 2 needed a custom rule manually defined by the user. After finishing that, the user could — with a little effort in searching the right snapshots — monitor the process of exporting a PDF. The hot spots of this process are the actual generation (about 65%) and the communication with the CC-jar (about 12,5%). It is possible to define a custom business transaction for the process mentioned above.

5.2.5 APP-UC-05 - CD-Invalidation

Vendor 1

In this scenario, Vendor 1 no errros were found because of nonexistent performance variations or problems. In general Vendor 1 was able to detected the CDB-Invalidation automatically.

Vendor 2

Due to missing erros and / or performance problems within this use case, accordingly there was no problem detected. It has to be mentioned that Vendor 2 automatically detected the CDB-Invalidation and, furthermore, marked it as a business transaction.

5.2.6 APP-UC - Conclusion

For the first use case: **APP-UC-01 - Application Restart**, neither of both solutions recognized, that the application restarted. As the vendors said, it is possible to detect a restart by setting an attribute (Vendor 1), or a flag (Vendor 2). At both solutions the agents stopped gathering data.
The second use case: **APP-UC-02 - Clickpath of a configuration**, goes to Vendor 2 . While with Vendor 1 's solution the user needs to start a transaction trace, Vendor 2 ' solution provides a wide range of information regarding the clickpath of an user.
In use case 3: **APP-UC-03 - Buildability Check**, as well as in use case 4: **APP-UC-04 - PDF Export**, both vendors came nearly to the same results. For these use cases Vendor 1 needs to start a transaction trace again, and catch or filter the slow traces. Vendor 1 demonstrated the Vendor 1 CA Synthetic Center, where you could see the loading time of every single image needed for the PDF-File.
In the last use case: **APP-UC-05 - CDB-Invalidation**, the outcome of both solutions was the same. Neither of both solutions found any errors and both of them detected the CDB-Invalidation automatically.

Chapter 6

Results & Overview

This Chapter gives an overview of the APM use case assessments. The following table shows each single use case assessment, the winner of each use case as well as the average.

6.1 Assessment Overview

The following table shows an aggregated view of all assessment results as well as the final overall result.

Use-Case		**Vendor 1**	**Vendor 2**	**Winner**
Installation	Installation	-1	0	Vendor 2
	User Management	0	2	Vendor 2
	Integration / Extensibilty	0	0	-
Configuration	Status-Quo Identification	0	0	-
	Runtime Application Architecture	0	2	Vendor 2
	Visualization / Dashboards	-3	3	Vendor 2
	Agent Configuration	0	0	-
Daily use	Hardware Monitoring	0	1	Vendor 2
	Root Cause Analysis / Deep Diving	-1	0	Vendor 2
	Analytics	0	1	Vendor 2
	Pro-Active Monitoring	0	1	Vendor 2
	Automation	0	0	-
Scala-bility	Scalability	-2	-1	Vendor 2
	Agent Functionality / Data Collection	0	1	Vendor 2
Privacy	Privacy	0	0	-

Average:	**-0,467**	**0,734**	**Vendor 2**

Explanation:

- ± 1 implies minimal differences
- ± 2 implies medium differences
- ± 3 implies major differences

6.2 Overall Conclusion

Below is a conclusion, summing up the results from a management view of the industry partners.

Functionality, Architecture, Integration

Vendor 1 as well as Vendor 2, both offer fully developed Application Performance Management Tools. Technically, there are no major differences.

The APM solution of Vendor 1 makes it a little bit easier to integrate the tool into the IT landscape of Industry Partner 1 and Industry Partner 2 . Mostly, because Vendor 1 provides existing interfaces, whereas with Vendor 2 the user has to modify some generic interfaces. Particulary in view of the end user management collector, Vendor 2 faces an architectural challenge due to calls, that have to be send from the controller server to this very end user management collector.

Usability

Usability is, where both tools really vary. The user interfaces and the intended target groups are the two points making the difference. During the evaluation, it did become clear, that Vendor 1 comes more from the expert tools (intended user group: developers and application responsibles), whereas Vendor 2 focuses also on business acountables.

Vendor 1 currently is moving away from the originally used Java UI (rich cient) to a modern web based (HTML5) user interface. Vendor 2 on the other hand, provides a continuous user experience; all information and controls are presented in the same web based user interface. This results in (much more) quicker and easier access to information when using the APM tool of Vendor 2 .

6.3 Recommendation

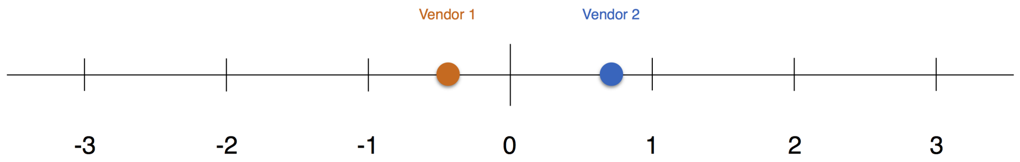


Figure 6.1. Average of assessments

Figure 6.1 shows the final result. Throughout the scope of the featured use cases, the tool of Vendor 2 established itself as the better tool with some advantages, compared to Vendor 1 . Therefore, we are recommending the tool of Vendor 2 .

Bibliography

[1] AppDynamics Documentation,
`https://docs.appdynamics.com`

[2] AppDynamics Datasheet Java,
`https://www.appdynamics.com/media/uploaded-files/1435166495/appd_ds_java-1.pdf`

[3] Appdynamics Blog: The Real Cost of Application Performance Management (APM) Ownership.,
`https://blog.appdynamics.com/apm/the-real-cost-of-application-performance-management-apm-ownership/`

[4] CA Documentation,
`https://docops.ca.com/ca-apm/10-1/en`

[5] CA Datasheet,
`http://www.ca.com/us/collateral/data-sheets/na/ca-application-performance-management.aspx`

[6] Gartner - Application Performance Monitoring,
`http://www.gartner.com/it-glossary/application-performance-monitoring-apm/`

[7] Gartner - Magic Quadrant for Application Performance Monitoring Suites,
`http://www.gartner.com/technology/reprints.do?id=1-2UD2RK1&ct=151217&st=sb`

[8] The Five Essential Elements of Application Performance Monitoring,
`http://www.realtimepublishers.com/book?id=168`

[9] APM Best Practices: Realizing Application Performance Management (Books for Professionals by Professionals),
Author: Michael J. Sydor
Publisher: Apress; 2010 edition (22 Dec. 2010)

Decleration

We declare that this thesis is the solely effort of the authors. We did not use any other sources and references than the listed ones. We have marked all contained direct or indirect statements from other sources as such. Neither this work nor significant parts of it were part of another review process. We did not publish this work partially or completely yet. The electronic copy is consistent with all submitted copies.

____

place, date, signature

____

place, date, signature

____

place, date, signature